Bulletin of the Seismological Society of America, Vol. 70, No. 1, pp. 79–99, February 1980

SOLVING TWO-POINT SEISMIC-RAY TRACING PROBLEMS IN A HETEROGENEOUS MEDIUM

Part 1. A General Adaptive Finite Difference Method

BY V. PEREYRA, W. H. K. LEE, AND H. B. KELLER

ABSTRACT

A study of two-point seismic-ray tracing problems in a heterogeneous isotropic medium and how to solve them numerically will be presented in a series of papers. In this Part 1, it is shown how a variety of two-point seismic-ray tracing problems can be formulated mathematically as systems of first-order nonlinear ordinary differential equations subject to nonlinear boundary conditions. A general numerical method to solve such systems in general is presented and a computer program based upon it is described. High accuracy and efficiency are achieved by using variable order finite difference methods on nonuniform meshes which are selected automatically by the program as the computation proceeds. The variable mesh technique adapts itself to the particular problem at hand, producing more detailed computations where they are needed, as in tracing highly curved seismic rays.

A complete package of programs has been produced which use this method to solve two- and three-dimensional ray-tracing problems for continuous or piecewise continuous media, with the velocity of propagation given either analytically or only at a finite number of points. These programs are all based on the same core program, PASVA3, and therefore provide a compact and flexible tool for attacking ray-tracing problems in seismology.

In Part 2 of this work, the numerical method is applied to two- and three-dimensional velocity models, including models with jump discontinuities across interfaces.

1. INTRODUCTION

Previous literature. In order to study the heterogeneous structure of the Earth, seismologists have developed several techniques to trace seismic rays. For example, Jackson (1970), Jacob (1970), Julian (1969, 1970), and Wesson (1970, 1971) developed numerical techniques to trace seismic rays in an inhomogeneous medium and applied them to study a variety of interesting problems in seismology. The first three authors formulated seismic-ray tracing as an initial value problem, and the last author presented both initial value and boundary value formulations.

As Wesson (1971, p. 741) pointed out, the tracing of seismic rays between two end points is required in seismological applications such as earthquake location (e.g., Engdahl and Lee, 1976) and the determination of three-dimensional velocity structure under a seismic array (e.g., Aki and Lee, 1976; Aki *et al.*, 1977). A common approach is to solve a series of initial value problems from one end point, and to iteratively seek the ray that passes through the other end point. Another approach is to solve the two-point boundary value problem directly. Wesson (1970, 1971), discussed the merits of both approaches and more recently Julian and Gubbins (1977) presented a comparison of these two methods, concluding that the boundary value problem approach is computationally faster.

Both Wesson (1970, 1971) and Julian and Gubbins (1977) used similar central finite difference approximations to the second order differential equations in solving the two-point seismic-ray tracing problem. Chander (1975) used a method due

originally to L. Euler which approximates the integral for the travel time by a sum and solves for the minimum time path directly. Yang and Lee (1976) showed that this formulation is equivalent to the central-difference approximation used by Wesson (1970, 1971).

A different approach for solving two-point seismic-ray tracing problems has been introduced by Yang and Lee (1976). They first reduced the second-order ray equations to a set of first-order equations, and then solved them using an adaptive finite-difference program written by Lentini and Pereyra (1975). For two-dimensional models where the velocity is a linear function of the coordinates, Yang and Lee (1976) showed that this approach gave more accurate solutions and used less computer time than the central-difference approach. This result is not surprising because considerable advances have been made by mathematicians in solving general two-point boundary problems [e.g., Bailey *et al.* (1968); Bellman and Kalaba (1965); Fox (1957); Keller (1968, 1974); Lentini and Pereyra (1974, 1975, 1977); Pereyra, (1967, 1968, 1973); Roberts and Shipman (1972)]. Very accurate and efficient methods have been developed, and the method of Lentini and Pereyra (1975) and its successor PASVA3 which is described here represent state-of-the-art techniques.

Plan of this paper. The present collaboration of two mathematicians and a seismologist is intended to provide a study of two-point seismic-ray tracing problems and to develop accurate and efficient computer programs to solve them. In Part 1 of this paper a general numerical method for solving ray-tracing problems will be described, and its use in a computer program will be discussed. Actually, this program is a general solver for a system of nonlinear first-order differential equations, and has been used in many other applications (Lentini and Pereyra, (1974, 1975, 1977). However, the numerical method will be described in an elementary manner, and its relationship to the problem of two-point seismic-ray tracing will be shown. In Part 2 of this paper, this numerical method will be applied to two- and three-dimensional velocity models, including models with jump discontinuities across interfaces.

In section 2, the equations governing the ray path in a two-point seismic-ray tracing problem are derived and it is shown how these equations may be reduced to a set of first-order equations.

The new features of the present numerical method are described in sections 3 through 5. High accuracy and efficiency are achieved by using variable-order finite difference methods on nonuniform meshes which are selected automatically by the program as the computation proceeds. The method requires the solution of large, sparse systems of nonlinear equations, for which a fairly elaborate iterative procedure has been designed. This in turn requires a special linear equation solver which takes into account the structure of the resulting matrix of coefficients.

The variable mesh technique allows the program to "adapt itself" to the particular problem at hand, and thus it produces more detailed computations where they are needed, as in the case of highly curved rays.

Section 6 shows how to deal with a heterogeneous medium in which the seismic velocity has discontinuities [see also Keller (1964)].

2. The General Two-Point Ray-Tracing Problem

Ray equations. Equations for the general two-point seismic-ray tracing problem in a heterogeneous and isotropic medium may be derived from Fermat's principle. If we use arc length, s, along the ray as the independent variable, then the differential

equation that the ray must satisfy is

$$\frac{d}{ds}\left(u(\eta)\frac{d}{ds}\eta\right) - \nabla u = 0, \tag{2.1}$$

where

$$\eta = (x, y, z), \quad \text{and} \quad u(\eta) = 1/v(\eta),$$

with $v(\eta)$ the velocity of propagation in the medium. Finally $\nabla u = (\partial u/\partial x, \partial u/\partial y, \partial u/\partial z)$ is the gradient of u. (All column vectors will be displayed as row vectors to save space.)

Equation (2.1) may be written as a set of three second-order nonlinear differential equations. Naturally, we have the additional constraint

$$\dot{x}^2 + \dot{y}^2 + \dot{z}^2 = 1, \tag{2.2}$$

where the dot denotes differentiation with respect to s.

The two-point ray-tracing problem consists of finding a solution of (2.1) and (2.2) that passes through two given points $P_0 = (x_0, y_0, z_0)$ and $P_1 = (x_1, y_1, z_1)$.

First it is shown that it is enough to consider equations (2.1); i.e., if an appropriate initial condition is imposed then equation (2.2) will automatically be satisfied. It turns out that such an additional condition can be imposed because the total arc length of the ray between P_0 and P_1 is also unknown.

Let us first expand equation (2.1):

$$\langle \nabla u, \dot{\eta}\rangle\dot{\eta} + u\ddot{\eta} - \nabla u = 0, \tag{2.3}$$

where $\langle\ ,\ \rangle$ denotes the inner product of two vectors. if we now take the inner product of (2.3) with $\dot{\eta}$ and observe that $\langle \dot{\eta}, \dot{\eta}\rangle = \|\dot{\eta}\|_2^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$, we obtain

$$\langle \nabla u, \dot{\eta}\rangle(\|\dot{\eta}\|_2^2 - 1) + u\langle \ddot{\eta}, \dot{\eta}\rangle = 0. \tag{2.4}$$

If we call $\zeta \equiv \|\dot{\eta}\|_2^2 - 1$ and observe that $\langle \ddot{\eta}, \dot{\eta}\rangle = \frac{1}{2}\, d/ds(\|\dot{\eta}\|_2^2)$,

then

$$\dot{\zeta} = -2v\langle \nabla u, \dot{\eta}\rangle\zeta. \tag{2.5}$$

Therefore, for any solution η of equations (2.1) we can write $\dot{x}^2 + \dot{y}^2 + \dot{z}^2 - 1 \equiv \zeta(s) = \zeta_0 \exp[-2]\int_{s_0}^{s} v\langle \nabla u, \dot{\eta}\rangle\, ds$, and if we choose $\zeta_0 = 0$, then $\zeta(s)$ will be identically zero and the constraint (2.2) will be automatically satisfied.

Writing (2.3) in detail leads to the following formulation for the two-point ray-tracing problem:

$$\begin{aligned} \ddot{x} &= v(-G(\eta)\dot{x} + u_x), \\ \ddot{y} &= v(-G(\eta)\dot{y} + u_y), \\ \ddot{z} &= v(-G(\eta)\dot{z} + u_z), \end{aligned} \tag{2.6}$$

where $G(\eta) = u_x\dot{x} + u_y\dot{y} + u_z\dot{z}$. If S is the total length of the ray between P_0 and P_1, then the boundary conditions are

$$\begin{aligned} x(0) &= x_0, \quad y(0) = y_0, \quad z(0) = z_0, \\ x(S) &= x_1, \quad y(S) = y_1, \quad z(S) = z_1, \end{aligned} \tag{2.7}$$

plus the nonlinear condition we deduced earlier

$$\dot{x}(0)^2 + \dot{y}(0)^2 + \dot{z}(0)^2 = 1. \tag{2.8}$$

Since S is unknown, the number of conditions is appropriate.

This formulation of the three-dimensional ray-tracing problem is similar to the one used by Julian and Gubbins (1977), but not identical. The main advantage of using an arc length parameterization lies in the fact that the functions $x(s)$, $y(s)$, and $z(s)$ are single valued, even when rays curve back on themselves.

Reduction to first-order systems. In sections 3 through 5 we describe a powerful algorithm for solving general systems of the form

$$\omega' = \boldsymbol{f}(\tau, \omega), \qquad \tau \in [0, 1], \tag{2.9}$$

subject to the nonlinear boundary conditions

$$\boldsymbol{g}[\omega(0), \omega(1)] = 0, \tag{2.10}$$

where ω, $\boldsymbol{f}$, $\boldsymbol{g}$ are vector functions of arbitrary dimension d. Therefore it will be convenient to express all our different ray-tracing problems in this standard format. Second or higher order systems of ordinary differential equations can be easily reduced to first-order systems by introducing auxiliary variables. For instance, system (2.6) reduces to

$$\begin{aligned} \dot{\omega}_1 &= \omega_2 \\ \dot{\omega}_2 &= v[-G(\omega)\omega_2 + u_x] \\ \dot{\omega}_3 &= \omega_4 \\ \dot{\omega}_4 &= v[-G(\omega)\omega_4 + u_y] \\ \dot{\omega}_5 &= \omega_6 \\ \dot{\omega}_6 &= v[-G(\omega)\omega_6 + u_z] \end{aligned}$$

where $G(\omega) = u_x\omega_2 + u_y\omega_4 + u_z\omega_6$, and the vector

$$\omega \equiv (x, \dot{x}, y, \dot{y}, z, \dot{z}).$$

Since in many of these calculations the travel time $T = \int_0^S u\, ds$ is of interest, a new variable, ω_7, is introduced to represent the partial travel time, and we also introduce a corresponding differential equation and initial condition

$$\dot{\omega}_7 = u, \qquad \omega_7(0) = 0.$$

With this addition the total travel time, $T = \omega_7(S)$ is computed to the same precision as the ray.

In order to determine the unknown parameter S, we make the change of variables $s \to \tau = s/S$, and introduce the trivial differential equation $S' = 0$, where now prime ($'$) will denote differentiation with respect to τ. Calling $\omega_8 \equiv (S)$, we obtain the final set of equations

$$\begin{aligned}
\omega_1' &= \omega_8\omega_2 \\
\omega_2' &= \omega_8 v(-G\omega_2 + u_x) \\
\omega_3' &= \omega_8\omega_4 \\
\omega_4' &= \omega_8 v(-G\omega_4 + u_y) \\
\omega_5' &= \omega_8\omega_6 \\
\omega_6' &= \omega_8 v(-G\omega_6 + u_z) \\
\omega_7' &= \omega_8 u \\
\omega_8' &= 0,
\end{aligned} \qquad \tau \in [0,1] \tag{2.11}$$

with the boundary conditions

$$\begin{aligned}
&\omega_1(0) = x_0, \quad \omega_3(0) = y_0, \quad \omega_5(0) = z_0, \quad \omega_7(0) = 0, \\
&\omega_1(1) = x_1, \quad \omega_3(1) = y_1, \quad \omega_5(1) = z_1, \\
&\omega_2^2(0) + \omega_4^2(0) + \omega_6^2(0) - 1 = 0.
\end{aligned} \tag{2.12}$$

Two-dimensional ray tracing. For two-dimensional problems, say in the (x, z) plane, it is possible to give a simpler formulation by choosing appropriate dependent variables.

Equations (2.1) for this two-dimensional case become

$$\begin{aligned}
\frac{d}{ds}\left(u\frac{dx}{ds}\right) - u_x &= 0, \\
\frac{d}{ds}\left(u\frac{dz}{ds}\right) - u_z &= 0, \\
\left(\frac{dx}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2 &= 1.
\end{aligned} \tag{2.13}$$

Introducing the new variable ψ defined by

$$\frac{dx}{ds} = \cos\psi, \qquad \frac{dz}{ds} = \sin\psi, \tag{2.14}$$

we find that

$$\ddot{x} = -\dot{z}\dot{\psi} \quad \text{and} \quad \ddot{z} = \dot{x}\dot{\psi}$$

so

$$\ddot{x}\dot{z} - \dot{x}\ddot{z} = -\dot{\psi}. \tag{2.15}$$

Expanding the first two equations in (2.13), multiplying respectively by $\dot{z}$ and $\dot{x}$, and substracting, we obtain

$$u(\ddot{x}\dot{z} - \ddot{z}\dot{x}) + \dot{x}u_z - \dot{z}u_x = 0.$$

Using (2.15), introducing the variables

$$\omega \equiv (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)^T \equiv (x, z, \psi, S, T)$$

and making the change of variables $s \to \tau = s/S$, the resulting first order system is

$$\begin{aligned}
\omega_1' &= \omega_4 \cos \omega_3 \\
\omega_2' &= \omega_4 \sin \omega_3, \qquad \tau \in [0, 1] \\
\omega_3' &= \omega_4 v(\omega_1, \omega_2)[u_z(\omega_1, \omega_2)\cos \omega_3 - u_x(\omega_1, \omega_2)\sin \omega_3] \\
\omega_4' &= 0 \\
\omega_5' &= \omega_4 u(\omega_1, \omega_2),
\end{aligned} \tag{2.16}$$

with the boundary conditions

$$\begin{aligned}
&\omega_1(0) = x_0, \qquad \omega_2(0) = z_0, \qquad \omega_5(0) = 0, \\
&\omega_1(1) = x_1, \qquad \omega_2(1) = z_1.
\end{aligned} \tag{2.17}$$

3. The Numerical Method

The numerical method used to solve equations of the form (2.9) to (2.10) (of which the ray tracing equations are examples) is based on a simple finite-difference approximation to $d\omega/d\tau$ on a mesh with $(J + 1)$ points in the interval $[0, 1]$. Consider a mesh π of points $\{\tau_j\}_{j=1,\cdots J+1}$ satisfying

$$0 = \tau_1 < \tau_2 < \cdots < \tau_{J+1} = 1 \tag{3.1}$$

and the trapezoidal rule approximation to equation (2.9)

$$\frac{\boldsymbol{W}_{j+1} - \boldsymbol{W}_j}{h_j} = \frac{1}{2}[\boldsymbol{f}(\tau_j, \boldsymbol{W}_j) + \boldsymbol{f}(\tau_{j+1}, \boldsymbol{W}_{j+1})], \; j = 1, \cdots J, \tag{3.2a}$$

with the boundary conditions

$$\boldsymbol{g}(\boldsymbol{W}_1, \boldsymbol{W}_{J+1}) = 0. \tag{3.2b}$$

Here the d-vectors $\boldsymbol{W}_j$ are meant to approximate $\omega^*(\tau_j)$, an exact solution of problem (2.9) to (2.10), and $h_j = \tau_{j+1} - \tau_j$ is the mesh spacing, which is *not* assumed to be uniform. This can be of importance if some component of the solution $\omega^*(\tau)$ varies rapidly in some subregion, since then the mesh can be made locally finer in order to

resolve this anomalous behavior in an efficient manner. Observe that this will be the case in ray tracing in regions where the path is highly curved.

Equations (3.2) form a system of $(J+1) \times d$ nonlinear algebraic equations in the same number of unknowns $\{W_{i,j}\}_{i=1,\ldots,d,j=1,\ldots,J+1}$, where $W_{i,j}$ is the ith component of $\boldsymbol{W}_j$. Using further vector notation equation (3.2) will be referred to as the *discrete* system and will be written as

$$\boldsymbol{F}_\pi(\boldsymbol{W}) = 0, \tag{3.3}$$

where

$$\boldsymbol{W} = \begin{bmatrix} W_{1,1} \\ W_{2,1} \\ \cdot \\ \cdot \\ \cdot \\ W_{d,1} \\ W_{1,2} \\ \cdot \\ \cdot \\ \cdot \\ W_{d,J+1} \end{bmatrix}, \qquad \boldsymbol{F}_\pi(\boldsymbol{W}) = \begin{bmatrix} \boldsymbol{g}^{(1)}(\boldsymbol{W}_1) \\ \boldsymbol{W}_2 - \boldsymbol{W}_1 - \dfrac{h_1}{2}(\boldsymbol{f}_1 + \boldsymbol{f}_2) \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{W}_{J+1} - \boldsymbol{W}_J - \dfrac{h_J}{2}(\boldsymbol{f}_{J+1} + \boldsymbol{f}_J) \\ \boldsymbol{g}^{(2)}(\boldsymbol{W}_1, \boldsymbol{W}_{J+1}) \\ \boldsymbol{g}^{(3)}(\boldsymbol{W}_{J+1}) \end{bmatrix} \begin{matrix} \\ \leftarrow d\text{-vectors} \\ \\ \\ \\ \\ \\ \end{matrix},$$

with $\boldsymbol{f}_j \equiv \boldsymbol{f}(\tau_j, \boldsymbol{W}_j)$. We have split the vector $\boldsymbol{g}$ of the boundary conditions into three subvectors $\boldsymbol{g} = (\boldsymbol{g}^{(1)}, \boldsymbol{g}^{(2)}, \boldsymbol{g}^{(3)})$, of dimensions p, r, and $q \equiv d - (p + r)$, representing the initial, coupled, and end conditions, respectively.

Under mild assumptions, system (3.3) will have an isolated solution, $\boldsymbol{W}^*$, near $\{\omega^*(\tau_j)\}$ provided $h \equiv \max_{j=1,\ldots,J} h_j$ is sufficiently small. Moreover, this discrete approximation will be accurate to order h^2. That is, there is a constant c such that

$$\| \boldsymbol{W}^* - \omega^* \| \equiv \max_{i=1,\ldots,d,j=1,\ldots,J+1} | W^*_{i,j} - \omega_i^*(\tau_j) | \leqq ch^2 \tag{3.4}$$

and $\boldsymbol{W}^*$ can be computed by a quadratically convergent Newton iteration if a sufficiently accurate starting trajectory W^0 is given (cf. Keller, 1974).

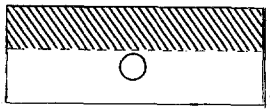
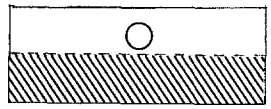
If we call $F_W(\boldsymbol{W})$ the Jacobian matrix of $\boldsymbol{F}_\pi$, we have that in $d \times d$ block form

$$F_W(\boldsymbol{W}) = \left(\frac{\partial \boldsymbol{F}_{\pi i}}{\partial \boldsymbol{W}_j} \right)_{i,j=1,\ldots,J+1}. \tag{3.5}$$

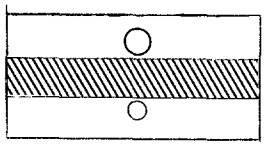
More specifically, $F_W(\boldsymbol{W})$ has the following block structure

$$F_W(\boldsymbol{W}) = \begin{bmatrix} A_1 & C_1 & \bigcirc & \cdot & \cdot & \bigcirc \\ B_2 & A_2 & C_2 & \bigcirc & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \bigcirc & \cdot \\ \cdot & & \cdot & \cdot & \cdot & \cdot \\ \cdot & \bigcirc & \cdot & B_J & A_J & C_J \\ D_1 & \bigcirc & \cdot & \cdot & B_{J+1} & A_{J+1} \end{bmatrix} \tag{3.6}$$

where the $d \times d$ subblocks C_j, B_j have the further sparseness indicated below

$B_j =$  $\} p$ $\qquad C_j =$  $\} p.$

The shaded regions indicate possible non-zero elements. Finally

$D_1 =$  $\} p$ $\} r$ $\} q.$

In order to construct this Jacobian matrix the program requires of the user the Jacobian matrix of the vector function $f(\tau, \omega)$ with respect to the variables ω, evaluated at all the grid points of the mesh π, and also the Jacobian matrices corresponding to the boundary conditions. Let us then define the $d \times d$ matrices of partial derivatives.

$$F_{W_j} = \left(\frac{\partial f_i}{\partial \omega_s}(\tau_j, W_j)\right)_{i,s=1,\cdots,d} ; \qquad j = 1, \cdots, J+1, \quad (3.7a)$$

and the matrices corresponding to the boundary conditions

$$G^{(1)}_{W_1} = \left[\frac{\partial g_i^{(1)}}{\partial W_{s,1}}(W_1)\right]_{i=1,\cdots,p,s=1,\cdots,d},$$

$$G^{(2)}_{W_1} = \left[\frac{\partial g_i^{(2)}(W_1, W_{J+1})}{\partial W_{s,1}}\right]_{i=1,\cdots,r,s=1,\cdots,d},$$

$$G^{(2)}_{W_{J+1}} = \left[\frac{\partial g_i^{(2)}(W_1, W_{J+1})}{\partial W_{s,J+1}}\right]_{i=1,\cdots,r,s=1,\cdots,d},$$

$$G^{(3)}_{W_{J+1}} = \left[\frac{\partial g_i^{(3)}(W_{J+1})}{\partial W_{s,J+1}}\right]_{i=1,\cdots,q,s=1,\cdots,d}. \quad (3.7b)$$

Then we have that the first p rows of A_1 are $G^{(1)}_{W_1}$, $D_1 \equiv G^{(2)}_{W_1}$, and the last $(r+q)$ rows of A_{J+1} are

$$\begin{pmatrix} G^{(2)}_{W_{J+1}} \\ G^{(3)}_{W_{J+1}} \end{pmatrix}.$$

An easy way of visualizing the rest of the matrix F_W is to think that block columns correspond to mesh points, while block rows correspond to equations. There is a little complication by the fact that the p initial conditions induce a shift of p rows on the whole matrix, and thus the partial derivatives corresponding to the jth difference equation appear as the last $(r+q)$ rows of block row j and the first p rows of block row $(j+1)$.

This ordering has been chosen because it puts F_W in almost block tridiagonal form; the only departure from this form is caused by D_1.

The necessary Jacobian matrices for the ray tracing problems (2.11) to (2.12) and (2.16) to (2.17) are given in the Appendix.

4. Solution of the Nonlinear Discrete Equations

The solution of (3.3) by Newton's method requires an initial approximation $\boldsymbol{W}^0$ and is then given by the iteration

a) For $\nu = 0, 1, \cdots$ solve the system of *linear* equations

$$\boldsymbol{F}_{\boldsymbol{W}}(\boldsymbol{W}^\nu)\Delta\boldsymbol{W}^\nu = -\boldsymbol{F}_\pi(\boldsymbol{W}^\nu). \tag{4.1a}$$

b) Correct to obtain a new iterate

$$\boldsymbol{W}^{\nu+1} = \boldsymbol{W}^\nu + \Delta\boldsymbol{W}^\nu. \tag{4.1b}$$

As we said before, if $\boldsymbol{W}^0$ is a sufficiently good initial estimate, this process will have the property that, for some constant k,

$$\| \boldsymbol{W}^{\nu+1} - \boldsymbol{W}^* \| \leqq k \| \boldsymbol{W}^\nu - \boldsymbol{W}^* \|^2, \tag{4.2}$$

where $\| \cdot \|$ stands for the infinity vector norm defined in (3.4); i.e., the convergence of the sequence $\{\boldsymbol{W}^\nu\}$ to the solution $\boldsymbol{W}^*$ of $\boldsymbol{F}_\pi(\boldsymbol{W}) = 0$ will be quadratic. It turns out that the norm of the error at the νth iteration is bounded by the norm of the residual

$$\| \boldsymbol{W}^\nu - \boldsymbol{W}^* \| \leq k' \| \boldsymbol{F}_\pi(\boldsymbol{W}^\nu) \|, \tag{4.3}$$

so it is enough to monitor this residual in order to obtain a satisfactory stopping criterion. Recalling that, after all, $\boldsymbol{W}^*$ is only an order h^2 approximation to the discretization of $\omega^*(t)$ in the mesh π [see (3.4)], then it will only be necessary to approximate $\boldsymbol{W}^*$ to a level compatible with this truncation error. Thus a reasonable criterion for the Newton iteration is to stop when the following inequality is satisfied

$$\| \boldsymbol{F}_\pi(\boldsymbol{W}^\nu) \| \leq \bar{k} h^2, \tag{4.4}$$

where $\bar{k}$ is a small constant.

A simple Newton iteration as indicated above may not be sufficient for difficult nonlinear problems for which a good initial estimate is not readily available. Our program incorporates some additional features which make the iterative process more robust and give the user some options which may be of help in difficult cases.

A way of enhancing the global convergence properties of Newton's method is to insist that the iteration have the property of *descent* with respect to an appropriate functional. We have borrowed for this purpose some techniques which are common in the unconstrained minimization of nonlinear functionals.

We consider instead of (4.1b) the following step-controlled correction procedure

$$\boldsymbol{W}^{\nu+1} = \boldsymbol{W}^\nu + \mu_\nu \Delta\boldsymbol{W}^\nu, \tag{4.1b'}$$

where $0 < \mu_\nu \leqq 1$ modifies the length of the Newton correction $\Delta \boldsymbol{W}^\nu$. Of course, $\mu_\nu = 1$ gives the quadratically convergent Newton iteration, but again, if we are not close enough to the desired solution $\boldsymbol{W}^*$, the process may diverge. In order to choose μ_ν so that convergence is induced in difficult cases we consider the auxiliary functional

$$r(\boldsymbol{W}) = \tfrac{1}{2} \| \boldsymbol{F}_\pi(\boldsymbol{W}) \|_2^2, \tag{4.5}$$

where $\| \cdot \|_2$ stands for the Euclidean norm of a vector, i.e., the square root of the sum of the squares of its components. The gradient of $r(\boldsymbol{W})$ is given by

$$\nabla r(\boldsymbol{W}) = \boldsymbol{F}_\pi^T \boldsymbol{F}_{\boldsymbol{W}}. \tag{4.6}$$

We shall say that the iteration (4.1a), (4.1b′) is of *descent* if

$$r(\boldsymbol{W}^{\nu+1}) \leqq c_\nu r(\boldsymbol{W}^\nu), \tag{4.7}$$

where $0 < c_\nu < 1$ will be specified later.

It is well known that the direction $-\nabla r(\boldsymbol{W})$ is such that the function $r(\boldsymbol{W})$ decreases the most rapidly along it, at least in a neighborhood of $\boldsymbol{W}$. This is the so-called direction of steepest descent. However, any direction $\boldsymbol{p}$ that forms an acute angle with $-\nabla r(\boldsymbol{W})$ will also be of descent. This condition is expressed by saying that the functional $r(\boldsymbol{W})$ will decrease locally along any direction $\boldsymbol{p}$ satisfying

$$\langle -\nabla r(\boldsymbol{W}), \boldsymbol{p} \rangle > 0, \tag{4.8}$$

where $\langle , \rangle$ denotes vector inner product. In fact, $\langle -\nabla r(\boldsymbol{W}), \boldsymbol{p} \rangle$ is a positive multiple of the cosine of the angle formed by the vectors $-\nabla r(\boldsymbol{W})$ and $\boldsymbol{p}$, and therefore (4.8) guarantees that this angle lies within $(-\pi/2, \pi/2)$.

It turns out that the Newton direction $\Delta \boldsymbol{W} = -\boldsymbol{F}_{\boldsymbol{W}}^{-1}(\boldsymbol{W}) \boldsymbol{F}_\pi(\boldsymbol{W})$ is always of descent for the functional $r(\boldsymbol{W})$, since

$$\begin{aligned} \langle -\nabla r(\boldsymbol{W}), \Delta \boldsymbol{W} \rangle &= \boldsymbol{F}_\pi^T(\boldsymbol{W}) \boldsymbol{F}_\pi(\boldsymbol{W}) \\ &= \| \boldsymbol{F}_\pi(\boldsymbol{W}) \|_2^2 = 2r(\boldsymbol{W}) > 0, \end{aligned} \tag{4.9}$$

and $r(\boldsymbol{W}) = 0$ only if $\boldsymbol{F}_\pi(\boldsymbol{W}) = 0$. This means that by choosing the step size μ_ν appropriately in the modified correction (4.1b′) it is always possible to satisfy a condition like (4.7). In fact, general results on iterative methods for unconstrained minimization guarantee that the following procedure due to Armijo [see Ortega and Rheinboldt (1970)] will always converge under appropriate assumptions.

Armijo's step control. Choose as μ_ν the first value of μ in the sequence $\{1, \frac{1}{2}, \frac{1}{4}, \cdots\}$ for which $r(\boldsymbol{W}^\nu) - r(\boldsymbol{W}^\nu + \mu \Delta \boldsymbol{W}^\nu) \geq \mu r(\boldsymbol{W}^\nu)$. From (4.1b′) and (4.7) we see then that $c_\nu = 1 - \mu$, and that in fact such a μ can always be found. A problem with this procedure is that in some instances it may produce a slowly convergent sequence by using very small steps, but that, in turn, is a sure indication that the problem is very difficult and that some auxiliary technique is called for.

Observing that the two sides of identity (4.9) are computed independently, equation (4.9) can be used to check the accuracy of the linear equation solver. In fact, the correction ΔW is obtained by solving the system of linear equations (4.1a),

and although a very stable algorithm is used, more of which will be discussed later, it is possible for the large matrix $F_W(W)$ to be ill conditioned. In such a case, the Newton correction may be very badly computed, up to the point that the identity (4.9) is not even nearly satisfied. Therefore we check both the descent property and the approximate verification of (4.9) and if either fails, we use the negative gradient direction $-\nabla r(W^\nu)$ instead of ΔW^ν in (4.1b′).

In performing step (4.1a) of the Newton iteration and also, as we shall see later, in computing global error estimates, it is necessary to solve linear systems of equations with a block quasitridiagonal matrix of coefficients of the form (3.6). Let us call for short $\mathcal{A} \equiv F_W(W)$.

A stable $\mathcal{L}\,\mathcal{U}$ factorization for this type of matrices is described in Keller (1974). An alternating partial pivoting strategy guarantees the stable construction of this decomposition with practically no fill-in, i.e., the sparse structure of $\mathcal{A}$ is preserved in the triangular factors $\mathcal{L}\,\mathcal{U}$ with the exception of the rows corresponding to D_1 [see (3.6)] in $\mathcal{L}$, which get filled.

5. Error Estimation, Adaptive Meshes, and Variable Order of Accuracy

As pointed out in section 3, the discretization (3.2) has order of accuracy h^2, even if a nonuniform mesh is used. Whenever there is *a priori* information on regions in which the solution $\omega^*(t)$ might have rapid variations it should be used to construct an appropriate mesh π. Rather than have the user worry about what is "appropriate," an automatic mesh selection procedure has been incorporated in this program which, in the course of the computation will try to find a good mesh for the problem. This is similar to what current state-of-the-art programs do in adaptive quadratures and in the solution of initial-value problems.

The order of accuracy of the basic approximation (3.2) will usually be too low and higher efficiency can be achieved by considering higher order formulas. On the other hand, if a direct approach to obtaining this higher order is made, the simple structure of (3.2) will be lost. The present approach to this problem is similar to the one used in the adaptive techniques mentioned above. A variable-order method based on deferred corrections has been developed (cf. Pereyra, 1967, 1968) which, coupled with the variable-mesh capabilities provides a fully adaptive tool for solving a wide variety of nonlinear two-point boundary value problems.

If equation (3.2) is written with W_j replaced by $\omega^*(\tau_j)$ and expand in Taylor's series around $\tau_j + h_j/2$, recalling that $f(\tau_j, \omega_j^*) \equiv \omega^*(\tau_j)$, the so-called *local truncation error* of the method is obtained.

$$\begin{aligned}\Phi_j &\equiv \frac{\omega_{j+1}^* - \omega_j^*}{h_j} - \frac{1}{2}[f(\tau_j, \omega_j^*) + f(\tau_{j+1}, \omega_{j+1}^*)] \\ &= -\frac{h_j^2}{12}\omega^{*'''}(\tau_j + h_j/2) + \mathcal{O}(h^4). \end{aligned} \tag{5.1}$$

Of course, further terms can be obtained by taking more terms in the Taylor expansion, but this will suffice for our present purposes.

A mesh π shall be called *equidistributing* if $\| \Phi_j \| =$ constant, $j = 1, \cdots, J$. Thus, roughly speaking, an equidistributing mesh will have small step sizes where the third derivative of the solution is large. A justification for the use of equidistributing meshes and an explanation on how to actually construct them can be found in Pereyra and Sewell (1975), and Lentini and Pereyra (1977). Here let it only be said

that we need in that process to approximate the leading term of the truncation error to order h^2, and that can be done by using the $\mathcal{O}(h^2)$ approximation $\boldsymbol{W}_j$. Obviously this will lead to a two-pass algorithm, in which an initial mesh π^0 is given and a discrete solution $\boldsymbol{W}_{\pi^0}$ is computed. Then $\boldsymbol{\Phi}_j$ is estimated and the mesh is corrected in an attempt to achieve equidistribution, and so on, until some stopping criterion is satisfied. This procedure has been incorporated into this program and on the average, only 2 or 3 passes are necessary to achieve an adequate level of equidistribution.

Although the extra computation required for the above procedure adds to the total cost of the computation, it turns out that there are at least two additional, very important uses for that information. Call $\boldsymbol{W}^0$ the computed $\mathcal{O}(h^2)$ solution to $\boldsymbol{F}_\pi(\boldsymbol{W}) = 0$, and $\boldsymbol{S}_1(\boldsymbol{W}^0)$ the $\mathcal{O}(h^2)$ approximation to the local truncation error $\boldsymbol{\Phi}$. Then, by solving the *linear* problem

$$\boldsymbol{F}_{\boldsymbol{W}}(\boldsymbol{W}^0)\boldsymbol{\Delta} = -\boldsymbol{S}_1(\boldsymbol{W}^0) \tag{5.2}$$

an $\mathcal{O}(h^2)$ approximation to the global error $\boldsymbol{W}_j^* - \omega^*(\tau_j)$ will be obtained, i.e.,

$$\boldsymbol{\Delta}_j = \boldsymbol{W}_j^* - \omega^*(\tau_j) + \mathcal{O}(h^2).$$

Solving equation (5.2) actually costs very little since the last available $\mathbb{L}\,\mathbb{U}$ decomposition of the Jacobian matrix can be used. If the mesh is adequate, this will usually be a very precise error estimate which is a feature presently lacking in most software.

But this is not all. It also turns out, that by solving the *nonlinear* problem

$$\boldsymbol{F}_\pi(\boldsymbol{W}) = \boldsymbol{S}_1(\boldsymbol{W}^0) \tag{5.3}$$

one obtains an $\mathcal{O}(h^4)$ approximate solution, i.e., if $\boldsymbol{W}^1$ is the computed solution of (5.3) then

$$\boldsymbol{W}_j^1 - \omega^*(\tau_j) = \mathcal{O}(h^4).$$

This is the first step in the deferred correction method. As a matter of fact, this process can be continued as long as the solution ω^* is sufficiently regular and as long as the mesh is adequate. Further terms in the expansion of $\boldsymbol{\Phi}$ must be approximated to increasing orders and then the kth correction will be accurate to order h^{2k+2}. This high-order accuracy allows one to solve problems very precisely on fairly coarse meshes. Since the cost of the computation increases with the number of mesh points, this is a very important feature. In comparison, the Julian and Gubbins (1977) method is only accurate to order h^2 (not h^3 as they erroneously indicate), and therefore it may require a large number of mesh points in order to achieve the accuracy required by the measured travel times.

All the systems to be solved will be of the form (5.3), i.e., like the original simple systems for the trapezoidal rule with a nonzero right hand side of the form $\boldsymbol{S}_k(\boldsymbol{W}^{k-1})$, which is a known vector. Thus the procedure explained in detail in section 4 is applicable to *all* the corrections

$$\boldsymbol{F}_\pi(\boldsymbol{W}) = \boldsymbol{S}_k(\boldsymbol{W}^{k-1}) \tag{5.4}$$

where $\boldsymbol{W}^{k-1}$ is the $\mathcal{O}(h^{2k})$ approximate solution after $(k-1)$ correction steps, and

$S_k(W^{k-1})$ is a finite difference approximation to the first k terms in the local truncation error expansion. For more details on this method, theoretical justification, and applications to other problems see Pereyra (1967, 1968, 1973) and Lentini and Pereyra (1974, 1977).

All these various techniques are arranged in a somewhat complex intertwined structure with a master control program that makes automatic decisions on when to refine the mesh, when to increase the order, and finally when to stop with a sufficiently accurate result (and corresponding error estimate), or an error message.

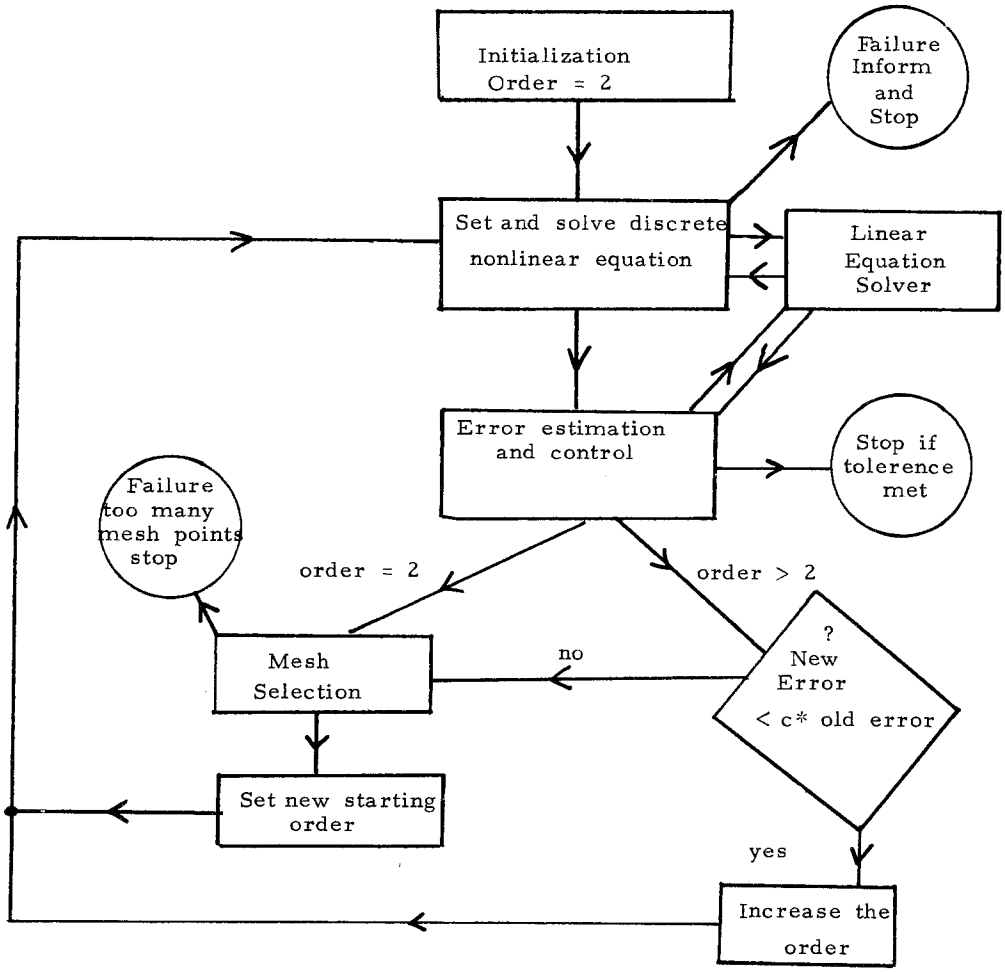


FIG. 1. Flow chart for the adaptive two-point boundary value problem solver.

A very schematic idea of this program is presented in the flow chart as shown in Figure 1.

It is assumed that the user request is for a discrete solution on a given mesh π^0 with absolute accuracy in all its components of size ϵ. Of course, a relative error tolerance, or a weighted error tolerance can be incorporated if that seems more suitable. Thus, the program will attempt to obtain $\bar{W}$ satisfying

$$\max \mid \bar{W}_{ij} - \omega_i^*(\tau_j) \mid \leqq \epsilon \tag{5.5}$$

on a mesh π^f containing the original mesh π^0, i.e., mesh refinements may occur.

After an initial check on the mesh to see if it requires refining, the basic strategy consists of trying to achieve equation (5.5) just by increasing the order of the method. There are several reasons for this strategy

1. The computational work involved in solving a nonlinear system like (5.4) is proportional to the number of mesh points J.
2. Once the first system $\boldsymbol{F}_{\pi^0}(\boldsymbol{W}) = 0$ is solved, all the remaining problems are small perturbations of it, and therefore considerable computational effort can be saved by keeping the Jacobian matrices and their $\mathbb{L}\,\mathbb{U}$ decompositions fixed all the time. Of course, after a mesh refinement the Jacobian matrix must be recomputed.
3. This procedure will converge linearly (a quasi or modified Newton iteration) because the Jacobian matrix depends on the ray path, but the rate of convergence is usually quite high, due to the accuracy of the approximate Jacobian and initial guess. In those circumstances, there is practically no difference between the number of iterations required by this modified algorithm and by the regular Newton method.

Unfortunately, unless the mesh π^0 is sufficiently fine to start with (with respect to the difficulty of the problem and the desired final accuracy ϵ), in general it won't be possible to achieve this goal by correcting only. After each correction the global error is estimated and compared with the error for the preceeding correction. If no substantial improvement has occurred then a mesh refinement is requested.

A number of error conditions guarantee that this process always terminates, either with a solution purportedly accurate to level ϵ, or with an indication of failure. Possible reasons for failure are

1. Error in some of the input parameters.
2. Divergence of Newton's method; this could occur if, for instance, the Jacobian matrix is very ill conditioned, and the safeguard mechanisms are not enough to steer the iteration away from this situation.
3. Not enough mesh points are available; this condition is of course computer dependent, i.e., the more storage that is available, the more mesh points can be used.
4. Too much accuracy is requested for the computer word length and the variations in scale of the solution to the problem.
5. There is no solution (shadow zone).

6. Treatment of Interfaces in the Medium

In ray tracing, as in many other applications, it is possible for problem (2.9) to have jump discontinuities within the domain of interest. These discontinuities appear in the function $\boldsymbol{f}(\tau, \omega)$.

If $\boldsymbol{f}$ is discontinuous with respect to the independent variable τ at a known location δ_1, $0 < \delta_1 < 1$, then the solution $\omega^*(\tau)$ or some of its derivatives may be discontinuous at δ_1. If $\boldsymbol{f}$ is discontinuous with respect to ω then the locations of the possible discontinuities in $\omega^*(\tau)$ or its derivatives are not known *a priori*.

Assume for simplicity that $\boldsymbol{f}$ becomes discontinuous when the ray, or trajectory $\omega^*(\tau)$, traverses a given surface $\varphi_1(\tau, \omega) = 0$ in $(0, 1) \times \mathbb{R}^d$, and that this occurs only once for $\tau \in (0, 1)$. Thus the condition for $\omega^*(\tau)$ to lie on this surface is that $\varphi_1[\tau, \omega^*(\tau)] = 0$. Then, call δ_1 the (unknown) value of τ for which the ray touches the interface.

Assuming that the desired solution has a known behavior across the discontinuity, i.e., obeys a jump discontinuity condition of the form $D[\omega^*(\delta_{1_-}), \omega^*(\delta_{1_+})] = 0$,

where D is d-dimensional and $\pm$ represent the values on the right and left of the discontinuity, this problem can be reduced to the form (2.9) with smooth data. Any number of interfaces of this type can be treated simultaneously in the same way, provided one knows the order in which they are traversed and to which region the ray enters (i.e., which velocity function applies). Observe that one can as easily consider reflected as refracted rays; or change from P to S waves and vice versa.

The idea of the reduction is to consider system (2.9) independently in the two regions $[0, \delta_1]$, $[\delta_1, 1]$, and use the interface condition $D = 0$ to couple these two problems. In each one of the intervals the problem is smooth. In the case of known crossing time δ_1, a system twice the size of the original one is considered (i.e., one copy for each subinterval), adding the interface condition as the necessary d additional boundary conditions.

The case of unknown crossing time requires the introduction of an auxiliary dependent variable $\delta_1(\tau)$, which satisfies the trivial first-order differential equation $\delta_1'(\tau) = 0$. The additional boundary condition is now given by the definition of the interface and the requirement that ω^* lies on it for $\tau = \delta_1$: $\varphi_1[\delta_1, \omega^*(\delta_1)] = 0$. If we call $[0, \delta_1]$, $[\delta_1, 1]$ regions I and II, respectively, map each interval into $[0, 1]$ and call ω_I, ω_{II} the solutions on these subintervals, we have an augmented system of the form

$$\begin{aligned} \omega_I' &= \delta_1 \boldsymbol{f}(\tau, \omega_I), \\ \omega_{II}' &= (1 - \delta_1)\boldsymbol{f}(\tau, \omega_{II}), \qquad \tau \in [0, 1] \\ \delta_1' &= 0. \end{aligned} \tag{6.1a}$$

with boundary conditions

$$\begin{aligned} \boldsymbol{g}[\omega_I(0), \omega_{II}(1)] &= 0, \\ D[\omega_I(1), \omega_{II}(0)] &= 0, \\ \varphi_1[\delta_1(1), \omega_I(1)] &= 0. \end{aligned} \tag{6.1b}$$

Apparently the size of the problem has been greatly increased from d equations to $(2*d + 1)$ equations, or in the case of n interfaces to $(n + 1)*d + n$ equations. But the real dimension of the problem is $J*E$, E being the number of equations and J the number of mesh points. Having mapped all the subregions into $[0, 1]$, it turns out that a mesh with K subintervals in $[0, 1]$ gets copied in each subregion, and therefore solving the transformed problem with $(K + 1)$ points is equivalent to solving the original one with $(n*K) + 1$ points and therefore the size of the system of nonlinear equations remains effectively constant. In fact, this transformation amounts to a reordering (and perhaps scaling) of the original equations. A similar procedure has been considered by Itoh (1975, 1976).

It is possible to solve this type of problem directly, without these transformations, but that approach requires substantial changes in the basic program PASVA3, so it will be left for future development.

6.1 *Ray tracing on piecewise continuous media.* The theory just developed can now be applied to equations (2.16). For simplicity, start with the case in which there is only one discontinuity in the velocity field, i.e., when the ray crosses the curve $\varphi(x, z) = 0$ (see Figure 2).

Let the value of s at the crossing point be δ_1, and let $I = [0, \delta_1]$, $II = [\delta_1, S]$.

Indicating with subscripts I or II the quantities in the two regions, the following set of ten differential equations are obtained from (2.16) and the procedure described above

$$\begin{aligned}
\omega_{1,J}' &= \omega_{4,J} \cos \omega_{3,J} \\
\omega_{2,J}' &= \omega_{4,J} \sin \omega_{3,J} \\
\omega_{3,J}' &= \omega_{4,J} v_J [u_{Jz} \cos \omega_{3,J} - u_{Jx} \sin \omega_{3,J}], \qquad J = I, II, \\
\omega_{4,J}' &= 0 \\
\omega_{5,J}' &= \omega_{4,J} u_J \qquad \tau \in [0, 1] \qquad (6.2)
\end{aligned}$$

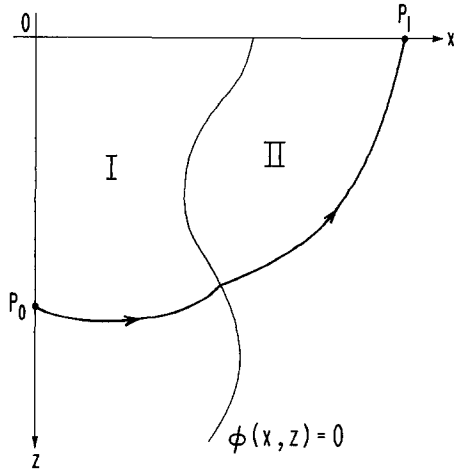


FIG. 2. Diagram to illustrate a velocity structure with one discontinuity.

with the boundary conditions

$$\begin{aligned}
\omega_{1_I}(0) &= x_0, \qquad \omega_{2_I}(0) = z_0, \qquad \omega_{5_I}(0) = \omega_{5_{II}}(0) = 0 \\
\omega_{1_I}(1) - \omega_{1_{II}}(0) &= 0, \qquad \omega_{2_I}(1) - \omega_{2_{II}}(0) = 0, \\
\omega_{1_{II}}(1) &= x_1, \qquad \omega_{2_{II}}(1) = z_1. \qquad (6.2a)
\end{aligned}$$

The remaining two conditions are given by Snell's Law, which the ray must obey when traversing a material discontinuity, and by the fact that the ray must be on the curve $\varphi = 0$ for $s = \delta_1$. Letting $P_{\delta_1} = [\omega_{1_I}(1), \omega_{2_I}(1)] = [\omega_{1_{II}}(0), \omega_{2_{II}}(0)]$ we get

$$\varphi(P_{\delta_1}) = 0,$$

$$\begin{aligned}
&v_{II}(P_{\delta_1})[\varphi_x(P_{\delta_1}) \sin \psi_I(1) - \varphi_z(P_{\delta_1}) \cos \psi_I(1)] \\
&\qquad - v_I(P_{\delta_1})[\varphi_x(P_{\delta_1}) \sin \psi_{II}(0) - \varphi_z(P_{\delta_1}) \cos \psi_{II}(0)] = 0. \qquad (6.2b)
\end{aligned}$$

For instance, for a vertical fault $\varphi(x, z) = x - x_F$ with x_F given, (6.2b) reduces to

$$\omega_{1_I}(1) = x_F$$

$$v_{II}(P_{\delta_1})\sin\psi_I(1) - v_I(P_{\delta_1})\sin\psi_{II}(0) = 0.$$

In general, we can use a similar technique to handle any number of interfaces. In order to keep the technical details simple we assume that we have k material interfaces $\varphi_i(x, y) = 0$, $i = 1, \cdots, k$, which subdivide the region of interest Ω into σ subregions in which we have different expressions for the velocity. We also assume that the ray traverses the interfaces and enters the various subregions in a given order. We simply write (6.2) with 2 replaced by $c + 1$, where c is the number of interfaces crossed. $\sum_{J=1}^{p} \omega_{4,J}(0) \equiv \delta_p$ gives the position of the pth crossing point, and $\sum_{J=1}^{c+1} \omega_{5,J}(1)$ gives the total travel time. The boundary conditions are similar to (6.2a)

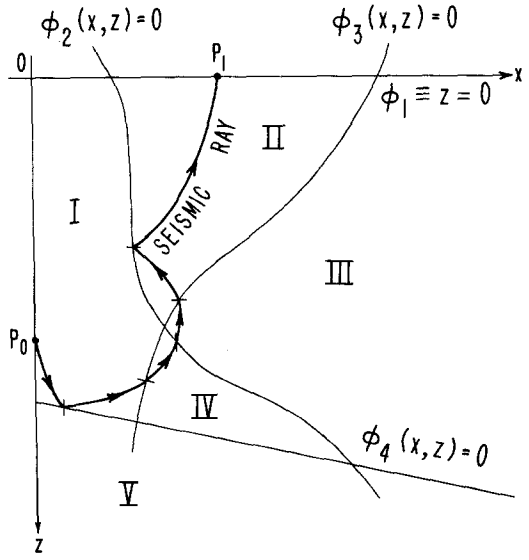


FIG. 3. Diagram to illustrate a complicated velocity structure ($c = 5$, $k = 4$, and $\sigma = 5$).

and we have c sets of conditions (6.2b). We have $5(c + 1)$ first-order differential equations and the same number of boundary conditions

4 end point conditions,
c initial time conditions,
$2c - 2$ continuity conditions,
$c - 1$ Snell's Law conditions,
$c - 1$ interface crossing conditions.

For some applications it may be of interest to replace some of the boundary conditions determining transmission across interfaces by reflection conditions. Thus a bounce against an interface also will be considered as a crossing. The method described here allows the treatment of fairly complicated structures (see Fig. 3) as will be shown in a later paper.

The procedure in three dimensions is the same, except that equations (2.11) to (2.12) are used.

ACKNOWLEDGMENTS

We wish to thank Dr. Bruce Julian of the U.S. Geological Survey for providing a copy of his paper with D. Gubbins on three-dimensional seismic ray tracing prior to publication, and to the referee for his very careful and critical reading.

The first author would like to acknowledge several very interesting conversations with Dr. J. A. Rial of the Caltech Seismological Laboratory.

This work was supported by U.S. Geological Survey Contract 14-08-0001-16777.

APPENDIX

Jacobian matrices for 2-dimensional ray tracing. The 5×5 Jacobian matrix of the right hand side of equations (2.16) is

$$J = \begin{bmatrix} 0 & 0 & -\omega_4 \sin \omega_3 & \cos \omega_3 & 0 \\ 0 & 0 & \omega_4 \cos \omega_3 & \sin \omega_3 & 0 \\ j_{31} & j_{32} & j_{33} & j_{34} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \omega_4 u_x & \omega_4 u_z & 0 & u & 0 \end{bmatrix}$$

with

$$j_{31} = \omega_4\{v_x(u_z \cos \omega_3 - u_x \sin \omega_3) + v(u_{zx} \cos \omega_3 - u_{xx} \sin \omega_3)\}$$

$$j_{32} = \omega_4\{v_z(u_z \cos \omega_3 - u_x \sin \omega_3) + v(u_{zz} \cos \omega_3 - u_{xz} \sin \omega_3)\}$$

$$j_{33} = \omega_4 v\{-u_z \sin \omega_3 - u_x \cos \omega_3\}$$

$$j_{34} = v\{u_z \cos \omega_3 - u_x \sin \omega_3\}.$$

In the piecewise continuous case, the only additional nontrivial derivatives stem from (6.2b). For the crossing of interface j we have, in an obvious notation, that the boundary condition is

$$B_j \equiv v_{j+1}[\varphi_{jx} \sin \psi_j(1) - \varphi_{jz} \cos \psi_j(1)] - v_j[\varphi_{jx} \sin \psi_{j+1}(0) - \varphi_{jz} \cos \psi_{j+1}(0)] = 0$$

and the relevant partial derivatives are

$$\frac{\partial B_j}{\partial x_{j+1}(0)} = v_{j+1,x}\{\varphi_{jx} \sin \psi_j(1) - \varphi_{jz} \cos \psi_j(1)\}$$
$$-v_j\{\varphi_{j,xx} \sin \psi_{j+1}(0) - \varphi_{j,zx} \cos \psi_{j+1}(0)\},$$

$$\frac{\partial B_j}{\partial z_{j+1}(0)} = v_{j+1,z}\{\varphi_{jx} \sin \psi_j(1) - \varphi_{jz} \cos \psi_j(1)\}$$
$$-v_j\{\varphi_{j,xz} \sin \psi_{J+1}(0) - \varphi_{j,zz} \cos \psi_{j+1}(0)\},$$

$$\frac{\partial B_j}{\partial \psi_{j+1}(0)} = -v_j\{\varphi_{jx} \cos \psi_{j+1}(0) + \varphi_{jz} \sin \psi_{j+1}(0)\},$$

$$\frac{\partial B_j}{\partial x_j(1)} = v_{j+1}\{\varphi_{j,xx} \sin \psi_j(1) - \varphi_{j,xz} \cos \psi_j(1)\}$$
$$-v_{jx}\{\varphi_{jx} \sin \psi_{j+1}(0) - \varphi_{jz} \cos \psi_{j+1}(0)\},$$

$$\frac{\partial B_j}{\partial z_j(1)} = v_{j+1}\{\varphi_{j,xz} \sin \psi_j(1) - \varphi_{j,zz} \cos \psi_j(1)\}$$
$$-v_{jz}\{\varphi_{jx} \sin \psi_{j+1}(0) - \varphi_{jz} \cos \psi_{j+1}(0)\},$$

$$\frac{\partial B_j}{\partial \psi_j(1)} = v_{j+1}\{\varphi_{jx}\cos\psi_j(1) + \varphi_{jz}\sin\psi_j(1)\}.$$

Jacobians for 3-dimensional ray tracing. The 8×8 Jacobian matrix corresponding to equations (2.11) has the nonzero $j_{\mu\nu}$ elements listed below. We will use the auxiliary quantities

$$G_{\omega_1} = u_{xx}\omega_2 + u_{xy}\omega_4 + u_{xz}\omega_6,$$
$$G_{\omega_2} = u_x,$$
$$G_{\omega_3} = u_{xy}\omega_2 + u_{yy}\omega_4 + u_{yz}\omega_6,$$
$$G_{\omega_4} = u_y,$$
$$G_{\omega_5} = u_{xz}\omega_2 + u_{yz}\omega_4 + u_{zz}\omega_6,$$
$$G_{\omega_6} = u_z.$$

Nonzero elements of Jacobian matrix

$$j_{12} = \omega_8, \qquad j_{18} = \omega_2;$$
$$j_{21} = \omega_8\{v_x(u_x - G\omega_2) + v(u_{xx} - G_{\omega_1}\omega_2)\},$$
$$j_{22} = -\omega_8 v(G_{\omega_2}\omega_2 + G),$$
$$j_{23} = \omega_8\{v_y(u_x - G\omega_2) + v(u_{xy} - G_{\omega_1}\omega_2)\}$$
$$j_{24} = -\omega_8 v\omega_2 G_{\omega_4}$$
$$j_{25} = \omega_8\{v_z(u_x - G\omega_2) + v(u_{xz} - G_{\omega_5}\omega_2)\}$$
$$j_{26} = -\omega_8 v G_{\omega_6}\omega_2$$
$$j_{28} = v(u_x - G\omega_2)$$
$$j_{34} = \omega_8, \qquad j_{38} = \omega_4;$$
$$j_{41} = \omega_8\{v_x(u_y - G\omega_4) + v(u_{xy} - G_{\omega_1}\omega_4)\},$$
$$j_{42} = -\omega_8 v G_{\omega_2}\omega_4,$$
$$j_{43} = \omega_8\{v_y(u_y - G\omega_4) + v(u_{xy} - G_{\omega_3}\omega_4)\},$$
$$j_{44} = -\omega_8 v(G_{\omega_4}\omega_4 + G),$$
$$j_{45} = \omega_8\{v_z(\omega_y - G\omega_4) + v(\omega_{yz} - G_{\omega_5}\omega_4)\},$$
$$j_{46} = -\omega_8 v G_{\omega_6}\omega_4, \qquad j_{48} = v(u_y - G\omega_4);$$
$$j_{56} = \omega_8, \qquad j_{58} = \omega_6;$$
$$j_{61} = \omega_8\{v_x(u_z - G\omega_6) + v(u_{xz} - G_{\omega_1}\omega_6)\},$$
$$j_{62} = -\omega_8 v G_{\omega_2}\omega_6,$$
$$j_{63} = \omega_8\{v_y(u_z - G\omega_6) + v(u_{yz} - G_{\omega_1}\omega_6)\},$$

$$j_{64} = -\omega_8 v G_{\omega_4}\omega_6,$$

$$j_{65} = -\omega_8\{v_z(u_z - G\omega_6) + v(u_{zz} - G_{\omega_5}\omega_6)\},$$

$$j_{66} = -\omega_8 v(G_{\omega_6}\omega_6 + G), \qquad j_{68} = v(u_z - G\omega_6);$$

$$j_{71} = \omega_8 u_x, \qquad j_{73} = \omega_8 u_y, \qquad j_{75} = \omega_8 u_z, \qquad j_{78} = u.$$

References

Aki, K., A. Christofferson, and E. S. Husebye (1977). Determination of the three-dimensional seismic structure of the lithosphere, *J. Geophys. Res.* **82,** 277–296.

Aki, K. and W. H. K. Lee (1976). Determination of three-dimensional velocity anomalies under a seismic array using first P-arrival times from local earthquakes. Part I—A homogeneous initial model, *J. Geophys Res.* **81,** 4391–4399.

Bailey, P. B., L. F. Shampine, and P. E. Waltman (1968). *Nonlinear Two Point Boundary Value Problems*, Academic Press, New York.

Bellman, R. and R. Kalaba (1965). *Quasilinearization and Nonlinear Boundary Value Problems*, American Elsevier, New York.

Chander, R. (1975). On tracing seismic rays with specified end points, *J. Geophys.* **41,** 173–177.

Engdahl, E. R. and W. H. K. Lee (1976). Relocation of local earthquakes by seismic ray tracing, *J. Geophys. Res.* **81,** 4400–4406.

Fox, L. (1957). *The Numerical Solution of Two Point Boundary Problems in Ordinary Differential Equations*, Clarendon Press, Oxford, 371 pp.

Itoh, K. (1975). On the numerical solvability of two point boundary value problems in a finite Chebyshev series for piecewise smooth differential systems, *Mem. Numer. Math.* **2,** 45–67.

Itoh, K. (1976). An a posteriori error estimation of approximate solutions of two point boundary value problems for piecewise smooth systems, *Mem. Fac. Sc., Kyushu Univ. Ser. A,* **30,** 75–94.

Jackson, P. L. (1970). Digital simulation of seismic waves, *Ph.D. Thesis*, Univ. Michigan, 84 pp.

Jacob, K. H. (1970). Three-dimensional seismic ray tracing in a laterally heterogeneous spherical earth, *J. Geophys. Res.* **75,** 6685–6689.

Julian, B. R. (1969). Regional variations in upper mantle structure beneath North America, *Ph.D. Thesis*, Calif. Inst. of Technology.

Julian, B. R. (1970). Ray tracing in arbitrarily heterogeneous media, *Technical Note, 1970-45*, Lincoln Lab., 17 pp.

Julian, B. R. and D. Gubbins (1977). Three dimensional seismic ray tracing, *J. Geophys.* **43,** 95–114.

Keller, H. B. (1964). Propagation of stress discontinuities in homogeneous elastic media, *SIAM Rev.* **6,** 356–382.

Keller, H. B. (1968). *Numerical Methods for Two Point Boundary Value Problems*, Blaisdell, London, 184 pp.

Keller, H. B. (1974). Accurate difference methods for nonlinear two point boundary value problems, *SIAM J. Numer. Anal.* **11,** 305–320.

Lentini, M. and V. Pereyra (1974). A variable order finite difference method for nonlinear multipoint boundary value problems, *Math. Comp.* **28,** 981–1004.

Lentini, M. and V. Pereyra (1975). An adaptive finite difference solver for non-linear two point boundary problems with mild boundary layers, *Comp. Sci. Dept. Stanford University Rept. STAN-CA-75-530*, 40 pp.

Lentini, M. and V. Pereyra (1977). An adaptive finite difference solver for nonlinear two point boundary problems with mild boundary layers, *SIAM J. Numer. Anal.* **14,** 91–111.

Ortega, J. M. and W. C. Rheinboldt (1970). *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 572 pp.

Pereyra, V. (1967). Iterated deferred corrections for nonlinear operator equations, *Numer. Math.* **10,** 316–323.

Pereyra, V. (1968). Iterated deferred corrections for nonlinear boundary value problems, *Numer. Math.* **11,** 111–125.

Pereyra, V. (1973). High order finite difference solution of differential equations, *Comp. Sci. Dept. Stanford University Rept. STAN-CS-73-348.*

Pereyra, V. and G. Sewell (1975). Mesh selection for discrete solution of boundary problems in ordinary differential equations, *Numer. Math.* **23,** 261–268.

Roberts, S. M. and J. S. Shipman (1972). *Two Point Boundary Value Problems: Shooting Methods*, American Elsevier, New York, 273 pp.

Wesson, R. L. (1970). Seismic ray computation in laterally inhomogeneous crustal models, *Ph.D. Thesis*, Dept. of Geophys., Stanford University.

Wesson, R. L. (1971). Travel-time inversion for laterally inhomogeneous crustal velocity models, *Bull. Seism. Soc. Am.* **61**, 729–746.

Yang, J. P. and W. H. K. Lee (1976). Preliminary investigations on computational methods for solving two point seismic ray tracing problems in a heterogeneous and isotropic medium. *U.S. Geol. Surv., Open-File Rept. 76-707*, 66 pp.

Departamento de Computacion
Universidad Central de Venezuela
Caracas 104, Venezuela (V.P.)

Applied Mathematics Department
California Institute of Technology
Pasadena, California 91125 (H.B.K.)

Office of Earthquake Studies
U.S. Geological Survey
Menlo Park, California 94025 (W.H.K.L.)

Manuscript received May 26, 1978